

WHAT IS CLAIMED IS

1. A system comprising:
 - a first agent;
 - a second agent connected to the first agent to receive and transmit events and data;
 - a processing agent to process a protocol, the processing agent being connected to the first agent, the processing agent being configured to send events to the first agent upon a change in the data being transmitted.
2. The system of claim 1 wherein the first agent is configured to monitor the data being transmitted to and received from the processing agent.
3. The system of claim 1 further comprising an event system coupled to the processing agent to store the events in the event system.
4. The system of claim 1 wherein the first agent includes an algorithm for flow control for the connections.

5. The system of claim 1 wherein the processing agent comprises a Secure Sockets Layer (SSL) system.

6. The system of claim 1 wherein the processing agent comprises a Server Load Balancing (SLB) system.

7. The system of claim 1 wherein the processing agent comprises an Extended Markup Language (XML) system.

8. The system of claim 1 wherein the events include at least one of an event type identification, a Transmission Control protocol (TCP) pointer, a controller handle, a controller length, and a controller prefetch.

9. The system of claim 1 wherein the data stored in the first agent includes a header and a data portion.

10. The system of claim 1 wherein the event system includes an event queue writer and event queue reader for the processing agent.

11. A method comprising:

transporting data between the first network agent and a second network agent through a processing agent, and

transporting events from a processing agent to the first agent upon a change in the data being transported.

12. The method of claim 11 wherein the first agent monitors data being transmitted to and received from the processing agent.

13. The method of claim 11 further comprising performing flow control of the data sent from the first agent to the second agent.

14. The method of claim 13 further comprising storing the events in an event system coupled to the processing agent.

15. The method of claim 11 wherein the first agent uses an algorithm for flow control for transporting data from the first agent through the processing agent to the second agent.

16. The method of claim 11 wherein the processing agent comprises a Secure Sockets Layer (SSL) System.

17. The method of claim 11 wherein the processing agent comprises a Server Load Balancing (SLB) system.

18. The method of claim 11 wherein the processing agent comprises an Extended Markup Language (XML) system.

19. The method of claim 11 wherein the events include at least one of an event type identification, a Transmission Control protocol (TCP) pointer, a controller handle, a controller length, and a controller prefetch.

20. The method of claim 11 wherein the data stored in the first agent includes a header and a data portion.

21. The method of claim 11 wherein the event system includes an event queue writer and event queue reader for the processing agent.

22. A machine-readable storage medium bearing machine-readable program code capable of causing a machine to:

- store data in a first agent;
- connect the first agent to a second agent to receive and transmit events;
- process a protocol by connecting a processing agent to the first agent, wherein the connections transport data between the first agent and the second agent and the

processing agent transports events to the first agent upon a change in the data being transmitted.

23. The system of claim 22 wherein the machine-readable program code further includes instructions to monitor the data being transmitted to and received from the processing agent.

24. The system of claim 22 wherein the processing agent is a Secure Sockets Layer (SSL) system.

25. The system of claim 22 wherein the processing agent is a Server Load Balancing (SLB) system.

26. The system of claim 22 wherein the processing agent is an Extended Markup Language (XML) system.

27. The system of claim 22 wherein the events include at least one of an event type identification, a Transmission Control protocol (TCP) pointer, a controller handle, a controller length, and a controller prefetch.

28. The system of claim 22 wherein the data stored in the first agent includes a header and a data portion.

29. The system of claim 22 wherein the event system includes an event queue writer and event queue reader for the processing agent.

30. A Transmission Control Protocol (TCP) processing system comprising:

a buffer to store data;

a first agent coupled to the buffer to receive and transmit events;

an event system coupled to the first agent to store the events in at least two event queues;

a first processing agent to process a protocol, the first processing agent having a first and a second connection with the first agent, wherein the first connection transports the data between the first agent and the first processing agent and the second connection transports the events between the first processing agent and the first agent; and

wherein the first agent is configured to monitor the data being transmitted to and received from the processing agent via the first and second connections.

31. The TCP processing system of claim 30 further comprising a second processing agent.

32. The TCP processing system of claim 30 wherein the processing agent is selected from a group comprising a Secure Sockets Layer (SSL) system, a Server Load Balancing (SLB) system, and an Extended Markup Language (XML) system.

33. The TCP processing system of claim 30 wherein the second processing agent is selected a group comprising a Secure Sockets Layer (SSL) system, a Server Load Balancing (SLB) system, and an Extended Markup Language (XML) system.

34. The TCP processing system of claim 30 wherein the protocol is selected from a group comprising a Secure Sockets Layer (SSL) protocol, a Server Load Balancing (SLB) protocol, and an Extended Markup Language (XML) protocol.

35. The TCP processing system of claim 30 wherein the first agent is configured to control the TCP receive window for performing flow control of the processing system.